HN

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/073,851	02/11/2002	Grzegorz J. Czajkowski	004-7056	8021

22120 7590 03/15/2005

ZAGORIN O'BRIEN GRAHAM LLP
7600B N. CAPITAL OF TEXAS HWY.
SUITE 350
AUSTIN, TX 78731

EXAMINER	
COURTENAY III, ST JOHN	
ART UNIT	**PAPER NUMBER**
2126	

DATE MAILED: 03/15/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 10/073,851	**Applicant(s)** CZAJKOWSKI ET AL.	
	Examiner St. John Courtenay III	**Art Unit** 2126	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 February 2002*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-25* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-25* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *11 February 2002* is/are: a)☒ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

ST. JOHN COURTENAY III
PRIMARY EXAMINER

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *Feb. 11, 2002*.

4) □ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

Detailed Action

1. Provisional Objection to claim 24:

Applicant is advised that should claim 1 be found allowable, claim 24 will be objected to under **37 CFR 1.75** as being a substantial duplicate thereof. When two claims in an application are duplicates or else are so close in content that they both cover the same thing, despite a slight difference in wording, it is proper after allowing one claim to object to the other as being a substantial duplicate of the allowed claim. See **MPEP § 706.03(k)**.

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claim 22 is rejected under 35 U.S.C. § 102(e) as being anticipated by **Blaukopf et al.** (U.S. Patent Application Publication US 2002/0095521).

As per independent claim 22:
Blaukopf teaches a computer system including a processor and a memory, the computer system comprising:

- a first process to execute a first software program coded in a safe language [e.g., see "first application is written in a platform independent language" and associated discussion §0016];

- a second process to execute a second software program coded in an unsafe language [e.g., see "second application 200 written in the processor's native code" and associated discussion §0016];

- an inter-process communication mechanism that allows data message communication between the first process and the second process [e.g., see §0017 "The first application 100 may pass a function call to the second application 200 through the first 120 and second 220 mediation modules, respectively; see cont'd discussion §0019, "Communication between mediation modules occurs using a stream protocol ..."; see cont'd discussion §0025: "Each such stream connection has its own memory buffers ..."];

- a first memory buffer object accessible by the first and the second process [e.g., see "first application is written in a platform independent language" and associated discussion §0016; see also discussion §0025: "Each such stream connection has its own memory buffers ..."]; and

- a second memory buffer object accessible by the first and the second process [e.g., see "second application 200 written in the processor's native code" and associated discussion §0016; see also discussion §0025: "Each such stream connection has its own memory buffers ..."].

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 23 and 25 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Blaukopf et al.** (U.S. Patent Application Publication US 2002/0095521) in view of **Chaney** et al. (European Patent Application EP 0 841 617 A2).

As per dependent claim 23:
Blaukopf discloses the invention substantially as claimed, as discussed above in the rejection of independent claim 22.

However, **Blaukopf** does not *explicitly* teach the following limitations:

Chaney teaches the first memory buffer object has a first address range in the first process, the second memory buffer object has a second address range in the first process and wherein the first address range and the second address range overlap [e.g., see col. 2, beginning line 40, i.e., "A technical feature of the invention is to use a single buffer for both request buffer space and response buffer space"].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to improve upon the system taught by **Blaukopf** by implementing the improvements detailed above because it would provide **Blaukopf's** system with the enhanced capability of "a system that makes more efficient use of the available buffer space and reduces the probability that a data packet will be delayed due to congestion in the queue buffers" [e.g., see Chaney, col. 1, lines 32-34].

As per independent claim 25:
Blaukopf teaches a method of processing a request to create a memory buffer object for use in a computer system, the method comprising:

- receiving a request (inherent) to create first and second memory buffer objects from a software program compiled to a computer system-independent language [e.g., see "first application is written in a platform independent language" and associated discussion §0016; see also §0017 "The first application 100 may pass a function call to the second application 200 through the first 120 and second 220 mediation modules, respectively; see cont'd discussion §0019, "Communication between mediation modules occurs using a stream protocol ..."; see cont'd discussion §0025: "Each such stream connection has its own memory buffers ..."];

However, **Blaukopf** does not *explicitly* teach the following limitations:

Chaney teaches allocating a first memory address range for the first memory buffer object in a first process executing the software program, allocating a second memory address range for the second memory buffer object in a first process executing the software program, the second memory address range at least partially overlapping the first memory address range; and [e.g., see col. 2, beginning line 40, i.e., "A technical feature of the invention is to use a single buffer for both request buffer space and response buffer space"].

Chaney teaches allocating a memory address range for each of the first memory buffer object and the second memory buffer object, in a second process, the second process executing native code [e.g., see col. 2, lines 41-48] .

It would have been obvious to one of ordinary skill in the art at the time the invention was made to improve upon the system taught by **Blaukopf** by implementing the improvements detailed above because it would provide **Blaukopf's** system with the enhanced capability of "a system that makes more efficient use of the available buffer space and reduces the probability that a data packet will be delayed due to congestion in the queue buffers" [e.g., see Chaney, col. 1, lines 32-34].

6. Claims 1-21, and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Blaukopf et al.** (U.S. Patent Application Publication US 2002/0095521) in view of **Minkoff et al.** (U.S. Patent 6,070,202).

As per independent claims 1 & 24:
Blaukopf discloses the invention substantially as claimed:

Blaukopf teaches a method for use with a computer system, the method comprising:

- providing a first software program compiled to platform-independent code for execution in a first process of the computer system [e.g., see "first application is written in a platform independent language" and associated discussion §0016];

- providing a second software program compiled to native code for execution in a second process of the computer system [e.g., see "second application 200 written in the processor's native code" and associated discussion §0016];

- sending a message from the first process to the second process to inherently request a memory buffer [e.g., see §0017 "The first application 100 may pass a function call to the second application 200 through the first 120 and second 220 mediation modules, respectively; see cont'd discussion §0019, "Communication between mediation modules occurs using a stream protocol ..."; see cont'd discussion §0025: "Each such stream connection has its own memory buffers ..."].

However, **Blaukopf** does not *explicitly* teach the following limitations:

Minkoff teaches sending a message from the first process to the second process to request a memory buffer [e.g., see allocation of a buffer in response to a memory request, col. 4, discussion beginning line 45].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to improve upon the system taught by **Blaukopf** by implementing the improvements detailed above because it would provide **Blaukopf's** system with the enhanced capability of a "memory management scheme that can better respond to the memory allocation needs of a particular operating environment [e.g., see Minkoff, col. 2, lines 39-40].

As per dependent claim 2:
Blaukopf inherently teaches requesting a first memory buffer in the first process, the first memory buffer having a first address range [e.g., see "memory buffers" discussion §0025, see §0017 "The first application 100 may pass a function call to the second application 200 through the first 120 and second 220 mediation modules, respectively; see cont'd discussion §0019, "Communication between mediation modules occurs using a stream protocol ..."; see cont'd discussion §0025: "Each such stream connection has its own memory buffers ..."; see also **Minkoff** allocation of a buffer in response to a memory request, col. 4, discussion beginning line 45].

As per dependent claim 3:
Blaukopf inherently teaches sending a message from the first process to the second process to request a second memory buffer in the second process [e.g., see §0017 "The first application 100 may pass a function call to the second application 200 through the first 120 and second 220 mediation modules, respectively; see cont'd discussion §0019, "Communication between mediation modules occurs using a stream protocol ..."; see cont'd discussion §0025: "Each such stream connection has its own memory buffers ..."; see also **Minkoff** allocation of a buffer in response to a memory request, col. 4, discussion beginning line 45].

As per dependent claim 4:
Blaukopf inherently teaches allocating a second address range in the second process for the second memory buffer [e.g., see see §0018 "The second application 200 may return a value, a series of values, or some other result; see cont'd discussion §0019, "Communication between mediation modules occurs using a stream protocol ..."; see cont'd discussion §0025: "Each such stream connection has its own memory buffers ..."; see

also **Minkoff** allocation of a buffer in response to a memory request, col. 4, discussion beginning line 45].

As per dependent claim 5:
Blaukopf teaches, in the second process, generating a first identifier associated with the second address range [e.g., see "In order to call a function, the second application first needs a reference to the function to be called" and associated discussion §0027; see discussion of second application 200 §0018; see also **Minkoff** allocation of a buffer in response to a memory request, col. 4, discussion beginning line 45].

As per dependent claim 6:
Blaukopf inherently creates a second memory buffer in the second process, the second memory buffer associated with the second address range [e.g., see §0018 "The second application 200 may return a value, a series of values, or some other result; see cont'd discussion §0019, "Communication between mediation modules occurs using a stream protocol ..."; see cont'd discussion §0025: "Each such stream connection has its own memory buffers ..."; see also **Minkoff** allocation of a buffer in response to a memory request, col. 4, discussion beginning line 45].

As per dependent claim 7:
Blaukopf inherently teaches recording information relating to the first memory buffer and to the second memory buffer [e.g., see §0017 "The first application 100 may pass a function call to the second application 200 through the first 120 and second 220 mediation modules, respectively; see cont'd discussion §0019, "Communication between mediation modules occurs using a stream protocol ..."; see cont'd discussion §0025: "Each such stream connection has its own memory buffers ..."; see also **Minkoff** allocation of a buffer in response to a memory request, col. 4, discussion beginning line 45].

As per dependent claim 8:
Blaukopf inherently teaches sending the first identifier and a second identifier from the second process to the first process, the second identifier representing the second memory buffer [e.g., see §0017 "The first application 100 may pass a function call to the second application 200 through the first 120 and second 220 mediation modules, respectively; see cont'd discussion §0019, "Communication between mediation modules occurs using a stream protocol ..."; see cont'd discussion §0025: "Each such stream connection has its own memory buffers ..."; see also **Minkoff** allocation of a buffer in response to a memory request, col. 4, discussion beginning line 45].

As per dependent claim 9:
Blaukopf inherently teaches mapping the first address range to a physical memory area identified by the first identifier [see first application 100 discussion §0017; ; see also **Minkoff** allocation of a buffer in response to a memory request, col. 4, discussion beginning line 45, block 520 as shown in fig. 5 and associated discussion col. 4, line 48].

As per dependent claim 10:
Blaukopf inherently teaches the first address range of the first process and the second address range of the second process both map to a common physical memory area [see memory buffers §0025 and physical memory disclosure §0012].

As per independent claim 11:
Blaukopf discloses the invention substantially as claimed:

Blaukopf teaches a method of processing a request to create a memory buffer object for use in a computer system, the method comprising:

- receiving an inherent request to create a memory buffer object from a software program compiled to a computer system-independent language [e.g., see §0017 "The first application 100 may pass a function call to the second application 200 through the first 120 and second 220 mediation modules, respectively; see cont'd discussion §0019, "Communication between mediation modules occurs using a stream protocol ..."; see cont'd discussion §0025: "Each such stream connection has its own memory buffers ..."];

- generating a first memory buffer object in a first process executing the software program [e.g., see "first application is written in a platform independent language" and associated discussion §0016; see also discussion §0025: "Each such stream connection has its own memory buffers ..."];

- generating a second memory buffer object via a second process, the second process executing native code [e.g., see "second application 200 written in the processor's native code" and associated discussion §0016; see also discussion §0025: "Each such stream connection has its own memory buffers ..."].

However, **Blaukopf** does not *explicitly* teach the following limitations:

Minkoff teaches sending a request to create a memory buffer [e.g., see allocation of a buffer in response to a memory request, col. 4, discussion beginning line 45].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to improve upon the system taught by **Blaukopf** by implementing the improvements detailed above because it would provide **Blaukopf's** system with the enhanced capability of a "memory management scheme that can better respond to the memory allocation needs of a particular operating environment" [e.g., see Minkoff, col. 2, lines 39-40].

As per dependent claim 12:
Blaukopf teaches the first memory buffer object and the second memory buffer object are mapped to a common memory area shared by the first process and the second process [e.g., see §0017 "The first application 100 may pass a function call to the second application 200 through the first 120 and second 220 mediation modules, respectively; see cont'd discussion §0019, "Communication between mediation modules occurs using a stream protocol ..."; see cont'd discussion §0025: "Each such stream connection has its own memory buffers ..."].

As per dependent claim 13:
Blaukopf teaches, at the first process, receiving an identifier associated with the second memory buffer object from the second process [e.g., see TCP/IP port numbers and associated discussion §0021; see also **Minkoff** allocation of a buffer in response to a memory request, col. 4, discussion beginning line 45].

As per dependent claim 14:
Blaukopf inherently teaches returning the identifier to the software program [e.g., see TCP/IP port numbers and associated discussion §0021; see also **Minkoff** allocation of a buffer in response to a memory request, col. 4, discussion beginning line 45].

As per dependent claim 15:
Blaukopf teaches the computer system is adapted to execute multiple software programs, a first of the software programs coded in a different programming language than a second of the software programs [e.g., see "first application 100" written in java and "second application 200" written in native code and associated discussion §0016].

As per independent claim 16:
Blaukopf discloses the invention substantially as claimed:

Blaukopf teaches a computer system including a processor and a memory, the computer system comprising:

- a first process to execute a first software program coded in a safe language [e.g., see "first application is written in a platform independent language" and associated discussion §0016];
- a second process to execute a second software program coded in an unsafe language [e.g., see "second application 200 written in the processor's native code" and associated discussion §0016]; and
- an inter-process communication mechanism [i.e., stream protocol §0019] that allows data message communication between the first process [first application 100, §0016] and the second process [second application 200, §0016], the inter-process communication mechanism *inherently* including a command that provides for transmission of a message from the first process to the second process to request creation of a direct buffer that is mapped from both the first process and the second process to a common

memory area [e.g., see §0017 "The first application 100 may pass a function call to the second application 200 through the first 120 and second 220 mediation modules, respectively; see cont'd discussion §0019, "Communication between mediation modules occurs using a stream protocol ..."; see cont'd discussion §0025: "Each such stream connection has its own memory buffers ..."].

However, **Blaukopf** does not *explicitly* teach the following limitations:

Minkoff teaches sending a message from the first process to the second process to request a memory buffer [e.g., see allocation of a buffer in response to a memory request, col. 4, discussion beginning line 45].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to improve upon the system taught by **Blaukopf** by implementing the improvements detailed above because it would provide **Blaukopf's** system with the enhanced capability of a "memory management scheme that can better respond to the memory allocation needs of a particular operating environment [e.g., see Minkoff, col. 2, lines 39-40].

As per independent claim 17:
Blaukopf discloses the invention substantially as claimed, as discussed above.

Blaukopf teaches a method for use with a computer system, the method comprising:

- providing a first software program compiled to platform-independent code for execution in a first process [e.g., see "first application is written in a platform independent language" and associated discussion §0016];

- providing a second software program compiled to native code for execution in a second process [e.g., see "second application 200 written in the processor's native code" and associated discussion §0016];

However, **Blaukopf** does not *explicitly* teach the following limitations:

Minkoff teaches requesting a first memory buffer in the first process, the first memory buffer having a first address range, sending a message from the first process to the second process to request a second memory buffer in the second process, and mapping the first address range to a physical memory area identified by a first identifier received from the second process. [e.g., see allocation of a buffer in response to a memory request, col. 4, discussion beginning line 45].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to improve upon the system taught by **Blaukopf** by implementing the improvements detailed above because it would provide **Blaukopf's** system with the enhanced capability of a "memory management scheme that can better respond to the memory allocation needs of a particular operating environment [e.g., see Minkoff, col. 2, lines 39-40].

As per independent claim 18:
Blaukopf discloses the invention substantially as claimed:

Blaukopf teaches a method for use with a computer system, the computer system having a first software program compiled to platform-independent code for execution in a first process [e.g., see "first application is written in a platform independent language" and associated discussion §0016] and having a second software program compiled to native code for execution in a second process [e.g., see "second application 200 written in the processor's native code" and associated discussion §0016].

However, **Blaukopf** does not *explicitly* teach the following limitations:

Minkoff teaches sending a message from the first process to the second process to request a memory buffer, allocating an address range in the second process for the memory buffer; and creating the memory buffer in the second process, the memory buffer associated with the address range [e.g., see allocation of a buffer in response to a memory request, col. 4, discussion beginning line 45].

It would have been obvious to one of ordinary skill in the art at the time the invention was made to improve upon the system taught by **Blaukopf** by implementing the improvements detailed above because it would provide **Blaukopf's** system with the enhanced capability of a "memory management scheme that can better respond to the memory allocation needs of a particular operating environment [e.g., see Minkoff, col. 2, lines 39-40].

As per dependent claim 19:
See the rejection of claim 5 above.

As per dependent claim 20:
See the rejection of claim 8 above.

As per dependent claim 21:
Minkoff teaches recording information relating to the memory buffer [e.g., see "metrics are also collected on buffers" and associated discussion col. 3, line 19] .

7. **Obviousness-type double patenting Rejection:**
The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See In re Goodman, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); In re Longi, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); In re Van Ornum, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); In re Vogel, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and In re Thorington,418 F.2d 528, 163 USPQ 644 (CCPA 1969).

> "Double patenting rejection of application claims was fully justified where applicant, in course of expanding first application to disclose enough more by way of details, alternatives, and additional uses to support broad, dominating, generic claims in later applications, has disclosed no additional invention or discovery other than that what was already claimed in patent on first application; there is significant difference between justifying broadening of claims and disclosing additional inventions." In re Van Ornum, 214 USPQ 761 (CCPA 1982).

8. Claims 1-25 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1 - 24 of parent application 09/841,719, now U.S. Patent 6,834,391 (**Czajkowski et al.**).

Although the conflicting claims are not identical, they are not patentably distinct from each other because of corresponding language that recites virtually all of the same elements and functions claimed in the previously patented invention, e.g., "a first process", "a second process", "native code", "platform independent code", and, in particular, inter-process communications between a process executing native code and a second process executing platform independent code.

The claimed differences would be obvious to a programmer of ordinary skill because the instant claims are merely broader and/or alternate variations of the claims recited in the parent application.

Because the instant claims merely eliminate and/or alternately claim limitations from the set of elements and functions claimed in the parent application, such modifications would be readily apparent to a programmer of ordinary skill.

Terminal Disclaimer

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b). Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

For post GATT applications, (i.e., applications filed after June 8, 1995), the rule § 1.321 (4) (c) (3) requires a provision that must be included. The following requirement is UNCHANGED by GATT and therefore a terminal disclaimer is required for the instant application, i.e., *"shall be enforceable only for and during such period that said patent is commonly owned with the application or patent which formed the basis for the rejection."*

9. **Prior Art not relied upon:**

Please refer to the references listed on the attached PTO-892 which are not relied upon in the claim rejections detailed above.

10. **Priority Claim Acknowledged**

Applicant's claim for priority under 35 U.S.C. § 119(e) with respect to provisional application 60/253,551, filed Nov. 28, 2000, is acknowledged.

How to Contact the Examiner:

Any inquiry concerning this communication or earlier communications from the examiner should be directed to St. John Courtenay III, whose telephone number is 571-272-3761. A voice mail service is also available at this number. The Examiner can normally be reached on Monday - Friday, 8:00 AM - 4:30 PM.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, An Meng-AI who can be reached on 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

All responses sent by U.S. Mail should be mailed to:

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

PTO CENTRAL FAX NUMBER:
703-872-9306

- Any inquiry of a general nature or relating to the status of this application should be directed to the **TC 2100 Group receptionist: (703) 305-3900.**

ST. JOHN COURTENAY III
PRIMARY EXAMINER